IPv6 on Linux servers

Dan Pritts
16 August 2013
Why IPv6?

• 7 Billion people

• 4 Billion IPv4 addresses ($2^{32}$)

• $340,282,366,920,938,000,000,000,000,000,000,000,000$ IPv6 addresses ($2^{128}$)
IPv4 Address Depletion

So What?

- NAT is horrendous
- Multi-layer NAT makes regular NAT look good
- “Polluted” address space (1.1.1.1?)
- Even if we at UM do not run out of addresses, much of the rest of the world will, and we need to talk to them.
- Microsoft paid $11.25/address in 2011
The network is changing

- Our choice is *not* whether we stick with the tried and true, or we migrate to IPv6.
Let’s Dive In

eth0  Link encap:Ethernet  HWaddr 52:54:00:F1:A7:B9
inet addr:141.211.255.68  Bcast:141.211.255.71
    Mask:255.255.255.248
    Scope:Global
inet6 addr: 2607:f018:704:ffff::68/64  Scope:Global
UP  BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
RX packets:45374 errors:0 dropped:0 overruns:0 frame:0
TX packets:14725 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:12914756 (12.3 MiB)  TX bytes:1799962 (1.7 MiB)
IPv4 address structure – 32 bits

An IPv4 address (dotted-decimal notation)

172 . 16 . 254 . 1

10101100 . 00010000 . 11111110 . 00000001

One byte = Eight bits

Thirty-two bits (4 x 8), or 4 bytes
IPv6 Address structure – 128 bits

An IPv6 address (in hexadecimal)

2001:0DB8:AC10:FE01:0000:0000:0000:0000

Zeroes can be omitted

2001:0DB8:AC10:FE01::

0010000000000001:0000110110111000:1010110000010000:1111111000000001:

0000000000000000:0000000000000000:0000000000000000:0000000000000000:0000000000000000
Real World Example

- 2607:f018:0704:ff00:0000:0000:0000:0068/64
- 2607:f018:704:ff00::68/64
What’s the /64?

• This is the subnet mask; it shows the number of address bits used for the local LAN vs. the rest of the Internet.
• Just like in IPv4
• 10.224.146.1/255.255.255.0
  10.224.146.1/24
• 141.211.255.68/255.255.255.248
  141.211.255.68/29
Static Addressing

- Some things don’t change much

inet6 addr: 2607:f018:704:ffff::68/64 Scope:Global
What about the broadcast?

- There isn’t one.
- Functions formerly done via broadcast are done via (more targeted) multicasts.
- You don’t have to configure broadcast or multicast addresses on interfaces.
- It’s an anachronism on IPv4 anyway. I’ve never used an IPv4 network where it couldn’t be computed from the address and netmask.
Link-local address

- Used for various low-level purposes
- Never routed off your subnet
- If everything is working you don’t need to worry about this

inet6 addr: fe80::5054:ff:feff1:a7b9/64 Scope:Link
2 Ways of Automatic Addressing

- **DHCPv6**
  - Much like the DHCP you are used to, but not supported on all OSes

- **Stateless address autoconfiguration (SLAAC)**
  - Lighter-weight; runs on routers, arguably lower security
  - Combination of router discovery and node auto-addressing (EUI-64 or privacy)
EUI-64 addressing

• My ethernet address is \textcolor{green}{52:54:00:F1:A7:B9}
• \textcolor{red}{2607:f018:704:ffff:5054:ff:fef1:a7b9/64}
• \textcolor{blue}{2607:f018:704:ffff:5054:00ff:fef1:a7b9/64}
• Flip the seventh bit of the first byte: 52->50 (local vs. global)
• \textcolor{red}{2607:f018:704:ffff:5054:00ff:fef1:a7b9/64}
Privacy Addresses

- Use a random 64-bit number
- Rotate periodically
- Privacy win; manageability lose
- On by default in windows client OSes
New messages in ICMPv6

- Router Solicitation/Advertisement (DHCP,RIP)
- ND - Neighbor Solicitation (ARP)
- MLD Multicast listener discovery (was IGMP)
Transition Strategies

- Dual Stack
  - Do both in parallel
  - Presumably what we will use at UM
- Backward Compatibility for IPv6-only networks
  - NAT64/DNS64
Tunneling & transition mechanisms

• There are various tunneling options for IPv4 users to get to the IPv6 internet. Don’t, except maybe to experiment.
  – Teredo, 6to4, 6rd, ISATAP
  – Large v6 servers might want to install 6to4 and Teredo gateways

• Happy Eyeballs
  – Apps and OS will prefer IPv6 if it exists.
  – User may have good IPv4 but broken IPv6 connectivity.
You said this talk would be about servers.
IP addressing on Red Hat & derivatives

- `/etc/sysconfig/network`
  NETWORKING_IPV6=true

- `/etc/sysconfig/network-scripts/ifcfg-eth0`
  IPV6INIT=yes
  IPV6ADDR=2607:f018:704:ffff::68
  # UMNet says to accept router announcements
  # this is how to do static
  IPV6_DEFAULTGW=2607:f018:704:ffff::2
  IPV6PREFIX=64
  IPV6_AUTOCONF=no  #EL6 bug?
IP addressing on Debian/Ubuntu

# The primary network interface
auto eth0
iface eth0 inet static
address 141.211.255.70
broadcast 141.211.255.71
netmask 255.255.248.0
gateway 141.211.255.65
iface eth0 inet6 static
address 2607:f018:704:ffff::70
netmask 64
# UMNet says to accept router announcements
# this is how to do static
gateway 2607:f018:704:ffff::2
Checking address, listeners, stats, routing

• ifconfig
• netstat –a [ –A inet6 ]
• netstat –r –A inet6
  – ip route command for IPv6?
• ip neighbor
  – standalone command like arp for ND?
Network toolkit

- Pretty robust; most open source in good shape
- ping6
- traceroute6
- mtr -6
- nmap -6
- socat
Security
Security

- Most things are similar; most of the same problems exist in v6 as v4.
- But tools are not always up to par (e.g., VFW)
- Feature Parity, or Feature Parody?
ip6tables

• Very similar to iptables
Too many addresses

- A /64 has way, way too many addresses to try to scan the whole thing
- IP-address based blacklisting (e.g., for spam) will not necessarily scale
- Wildcard reverse DNS (not)
Private IP addresses?

- There is no “private” IPv6 address space (like 10.0.0.0 or 192.168.0.0)*
- If you have business processes that use this space, think about how to transition
- OTOH IPv4 is not going away anytime soon

* “site-local” is defined, but is deprecated by RFC 3879
Applications
Apache httpd

- If you don’t configure up specific IP addresses, it Just Works
  
  Listen 80

- If you do configure up specific IPs, it’s still pretty easy.

<VirtualHost 141.211.255.68:443
  [2607:f018:704:ffff::68]:443>
Apache httpd

- Did you remember to open up port 80 in ip6tables?
BIND

• Again, pretty simple

listen-on-v6 port 53 { 2607:f018:704:ffff::68; };
BIND

- IPv6 info in the DNS
  - AAAA records instead of A records
  - ip6.arpa instead of in-addr.arpa.net

V6test0 A 141.211.255.68
V6test0 AAAA 2607:F018:704:ffff::68
BIND

- Nibble format for reverse lookups (ugh)

% host www.internet2.edu
webprod0.internet2.edu has IPv6 address 2001:48a8:68fe::151
% host 2001:48a8:68fe::151
1.5.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.e.f.8.6.8.a.8.4.1.0.0.2.ip6.arpa
domain name pointer www.internet2.edu.
BIND

2001:48a8:68fe:0000:0000:0000:0000:0151

Read it backwards...

1.5.1.0 . 0.0.0.0 . 0.0.0.0 . 0.0.0.0 . 0.0.0.0 .
0.0.0.0 . e.f.8.6 . 8.a.8.4 . 1.0.0.2
.ip6.arpa.
Q: How do I pre-fill an end-user subnet with reverse lookups?

A: You don’t.

How about just a dynamic answer? Code contributions welcome.

Dynamic DNS (yuck)

Stop requiring that clients have working reverse lookups.
BIND

• Did you remember to open up port 53 in ip6tables?
Others

• In general, bare IPv6 addresses OR addresses in brackets are accepted by most software.
• If it already uses :portnumber, try brackets.
The whole stack has to work

Case study: Internet2 video streaming

- End-user computer & campus network: check.
- WAN: check.
- Internet2 server LAN, DNS, etc: check.
- Internet2 web server OS & httpd: check.
- Internet2 video streaming server: check.
- Internet2’s video player applet? BZZZZZ!
Further reading

- Wikipedia articles are quite good
Thanks!

danno@umich.edu
734-615-1517
http://www-personal.umich.edu/~danno/slides/20130816/